International Journal of Computer Vision 30(1), 27–42 (1998)

The Tractability of Segmentation and Scene Analysis

MARTIN C. COOPER
IRIT, University of Toulouse III, 31062 Toulouse, France

Received March 4, 1997; Revised April 7, 1998; Accepted April 23, 1998

Abstract. One of the fundamental problems in computer vision is the segmentation of an image into semantically meaningful regions, based only on image characteristics. A single segmentation can be determined using a linear number of evaluations of a uniformity predicate. However, minimising the number of regions is shown to be an NP-complete problem. We also show that the variational approach to segmentation, based on minimising a criterion combining the overall variance of regions and the number of regions, also gives rise to an NP-complete problem.

When a library of object models is available, segmenting the image becomes a problem of scene analysis. A sufficient condition for the reconstruction of a 3D scene from a 2D image to be solvable in polynomial time is that the scene contains no cycles of mutually occluding objects and that no range information can be deduced from the image. It is known that relaxing the no cycles condition renders the problem NP-complete. We show that relaxing the no range information condition also produces an NP-complete problem.

Keywords: segmentation, scene analysis, computational complexity, NP-completeness

1. Tractability of Vision Problems

Segmentation is perhaps the most basic operation in low-level vision. Important applications are encountered in, among other areas, the analysis of satellite imagery, medical imagery, geology and metallurgy (Borisenko et al., 1987; Fu and Mui, 1981; Haralick and Shapiro, 1985; Pal and Pal, 1993). When a library of object models is available, the segmentation problem falls in the domain of scene analysis.

Different formal specifications of the segmentation problem exist. When choosing among them it is critical to know which problems are tractable and which are intractable. Apparently small changes in the problem specification can change a problem solvable in polynomial time into an NP-complete problem. The way in which a problem is expressed is often as important as the algorithm which is used to solve it. Establishing the borderline between those segmenation problems which are solvable in polynomial time and those which are intractable is clearly of great practical importance to computer vision engineers.

Before embarking on an analysis of the tractability of different versions of the segmentation problem, it is interesting to try to identify what makes certain vision problems easy and others intractable.

Certain problems are easy because they can be solved by application of local operations only. The detection of local features, such as edge segments, corners, discontinuities of curvature and zeros of curvature clearly fall into this category. In fact, this class of easy problems can be enlarged to include all problems of *unconstrained detection*. The identification and localisation of sets of local features which could correspond to partially visible rigid objects, occluded by unknown objects of arbitrary shape, is another example of unconstrained detection. This problem, although far from trivial, is solvable in polynomial time. If, for example, three local features in the image are sufficient to determine the three-dimensional position and orientation of an object, then an exhaustive search over all triples of pairs of matching local features in the image and the object model will solve it. The hypothesise and test approach to the recognition of partially occluded objects is based on the search for matching sets of local features (Ayache and Faugeras, 1986). An orthogonal approach to the same problem is an exhaustive search in transformation space (Cass, 1992).

To resume, unconstrained detection problems are tractable because the search space is not combinatorial.

What makes the interpretation of multi-object images intractable in the most general case is the presence of constraints between the interpretations of different parts of the image, meaning that the interpretation must be treated as a whole and cannot be decomposed into independent non-combinatorial subproblems. The most basic physical constraint that two solid objects cannot occupy the same physical space, and hence do not intersect, is sufficient to make the interpretation of multi-object images NP-complete (see Section 7).

The line drawing labelling problem is a classic example of a combinatorial problem in vision, since the label for one line may influence the possible labels for all other lines in the drawing. Huffman (1971) and Clowes [1971] gave constraints which must be satisfied by a semantic labelling of the lines in a drawing of polyhedra with trihedral vertices. Determining whether a drawing has a legal global labelling according to the Huffman-Clowes scheme has been shown to be NP-complete (Kirousis and Papadimitriou, 1988). Some drawings which can be labelled according to this scheme are not physically realisable (Sugihara, 1986), but the realisability problem has also been shown to be NP-complete (Kirousis and Papadimitriou, 1988).

We call the problem of reconstructing a three-dimensional scene from a two-dimensional image, subject to constraints concerning the physical realisability of the scene, *constrained reconstruction*.

If, in general, constrained reconstruction can give rise to intractable problems, this is not always the case. More restrictive (Dendris et al., 1994; Kirousis, 1990; Kirousis and Papadimitriou, 1988) or less restrictive (Cooper, 1997) versions of the Huffman-Clowes labelling scheme exist which have 0/1/all constraints, a class of tractable constraints (Cooper et al., 1994; Kirousis, 1993). A constraint satisfaction problem with 0/1/all constraints can be viewed as a generalisation of the well known tractable problem 2SAT to multi-valued logics, and is solvable in polynomial time. Knowledge of vanishing points in an image is also sufficient to convert the Huffman-Clowes labelling problem into a polynomially solvable problem (Parodi and Torre, 1994). This is again achieved by a reduction to 2SAT.

The interpretation of multi-object images also has an important tractable version. A single interpretation of an image composed of known flat objects occluding each other can be found in low-degree polynomial time, provided that the objects do not form cycles in depth. Furthermore, a data structure combining all valid interpretations (of which there may be a exponential number) can also be determined in low-degree polynomial time (Cooper, 1992).

A simplifying operation which can often convert an intractable problem into an easy problem is dimensionality reduction. A problem which is intractable on a two-dimensional image may become tractable on a one-dimensional image (a string). On the other hand, applying an optimisation criterion may render an otherwise tractable problem intractable. For example, in this paper we show that finding a single valid segmentation of a two-dimensional image becomes intractable when a global optimisation criterion is applied, and that the segmentation problem becomes easy again when an optimisation criterion is applied to a one-dimensional image.

In conclusion, whereas the unconstrained detection of features and sets of features is tractable, the reconstruction of a three-dimensional scene from a two-dimensional image subject to constraints on the physical realisability of the scene is, in general, an NP-hard problem. However, interesting tractable versions of the same problem often exist in which constraints have been either tightened or slackened. An optimisation criterion is an artificial global constraint which can render a tractable problem NP-hard.

2. Definitions

In the first part of this paper we study the problem of segmenting an image independently of any semantic information. The optimal segmentation found will almost always be an oversegmentation, in the sense that regions correspond to parts of objects rather than whole objects, due to the presence of surface-normal discontinuities, shadows, specular reflections or surface markings. On the other hand, constrast failure between two objects whose projections are adjacent in the image leads to the inevitable merging of two unrelated regions.

Thresholding methods (Lim and Lee, 1990) are clearly efficient but inadequate in many applications since they ignore the relative spatial disposition of pixels. Edge-based methods ignore the values of pixels which are not close to sharp discontinuities in the image. Both of these approaches are based on inherently local operations. A more global approach is to attempt to fit a piecewise smooth function to the image while simultaneously minimising the number of regions (Beaulieu and Goldberg, 1989) or the total length of region boundaries (Mumford and Shah, 1989; Nitzberg et al., 1991). We begin with a discrete version of the segmentation problem based on a uniformity predicate, before showing that the complexity results obtained can be generalised to the variational formulation of the optimal segmentation problem.

The following definition of a valid segmentation, based on a uniformity predicate for regions, is adapted from Pavlidas (1977).

Let X be a grid, on which is defined an image with pixel values $I(p)$ for all points $p \in X$. A one-dimensional grid is represented by

$$X = \{i : 1 \leq i \leq N\}$$

and a two-dimensional square grid by

$$X = \{(i, j) : 1 \leq i \leq N \text{ and } 1 \leq j \leq N\}$$

Definition 2.1. A *uniformity predicate* U assigns a truth value $U(Y)$ to each non-empty subset Y of the grid X as a function only of the values of $I(p)$ for $p \in Y$, and satisfies the subset property:

$$(Z \subseteq Y \wedge Z \neq \emptyset \wedge U(Y)) \Rightarrow U(Z)$$

Furthermore, $U(Y) =$ true for all $Y \subseteq X$ such that $|Y| = 1$.

Definition 2.2. A *segmentation* of X according to the uniformity predicate U is a partition of X into regions $X_1, \ldots, X_r$ such that

(i) each X_i is connected
(ii) $U(X_i) =$ true for $i = 1, \ldots, r$
(iii) $U(X_i \cup X_j) =$ false for all pairs of adjacent regions X_i, X_j.

Note that, by the definition of a partition, $X_1 \cup \cdots \cup X_r = X$ and $X_i \cap X_j = \emptyset$ for all $i \neq j$. Different definitions of a valid segmentation result from the different possible definitions of connected, such as 4-connected or 8-connected in two dimensional images. A region X_i is connected if between each pair of pixels $p, q \in X_i$, there is a path $p = p_1, p_2, \ldots, p_t = q$ of pixels such that, for each $i \in \{1, \ldots, t-1\}$, p_i and p_{i+1} are adjacent. In the definition of 4-connectedness, pixel (i, j) is only adjacent to pixels $(i-1, j)$, $(i+1, j)$, $(i, j-1)$, $(i, j+1)$, whereas in the definition of 8-connectedness, (i, j) is also adjacent to pixels $(i-1, j-1)$, $(i-1, j+1)$, $(i+1, j-1)$, $(i+1, j+1)$. In this paper we assume that connected means 4-connected. 8-connectedness provides unsatisfactory results when trying to minimise the number of regions in a segmentation. For example, a chessboard is composed of only two 8-connected regions, the black squares and the white squares.

An image may have an exponential number of valid segmentations. It is therefore natural to try to discriminate between them by means of some optimisation criterion.

Definition 2.3. A *minimal segmentation* of an image is such that the number of regions is minimal over all valid segmentations.

This criterion is derived from the assumption that simpler interpretations are more likely. Condition (iii) in Definition 2.2 only ensures that a segmentation is a partition of the image into coherent regions which *locally* minimises the number of regions, in that no two adjacent regions can be merged to form a coherent region.

To illustrate the utility of the global minimisation criterion, consider the image shown in Fig. 1(a) in the form of a matrix of grey levels. Assuming that U is the uniformity predicate

$$U(X) = \text{true} \quad \text{iff} \quad \text{for all points } p, q \in X \quad |I(p) - I(q)| \leq 1 \tag{1}$$

Figure 1(b) shows a minimal segmentation. It is clear that this segmentation is preferable to the valid segmentation in Fig. 1(c) which is non-optimal.

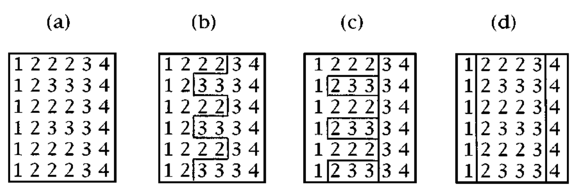

Figure 1. (a) An image, (b) a minimal segmentation, and (c), (d) two non-minimal segmentations.

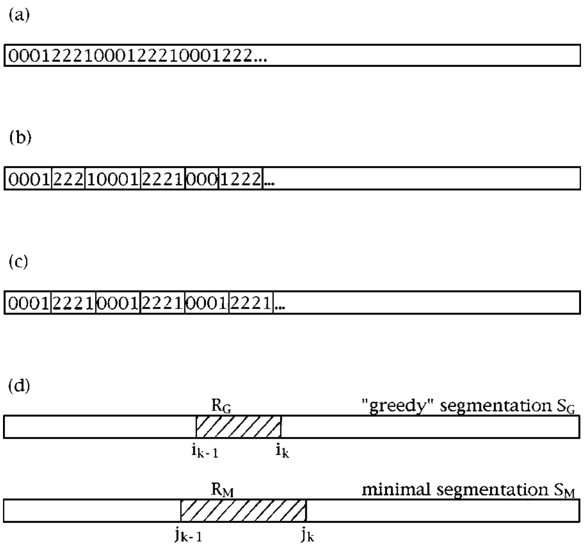

Figure 2. (a) A one-dimensional image, (b) a segmentation of this image, (c) the greedy segmentation, and (d) comparison of the greedy segmentation and a minimal segmentation.

Even in one dimension, an image can have an exponential number of valid segmentations, as the following example shows. Let U be the same simple uniformity predicate defined in (1). In a valid segmentation of the image in Fig. 2(a), sequences of 0's and sequences of 2's must be separated by a boundary, but the boundaries can lie either to the left or to the right of the 1. For example, one possible segmentation is given in Fig. 2(b), but there are clearly an exponential number of valid segmentations.

It should be noted that the number of minimal segmentations may also be exponential. This is the case in the one-dimensional example of Fig. 2, in which all valid segmentations (according to Definition 2.2) are minimal. It should also be noted that there is no guarantee that the correct segmentation will be minimal. For example, Fig. 1(d) shows a non-minimal segmentation of the image in Fig. 1(a). This segmentation is just as plausible as the minimal segmentation shown in Fig. 1(b).

In the presence of Gaussian noise, a simple threshold on the difference between pixel values of a region, as embodied by the uniformity predicate given by (1), is too naive as a means of distinguishing between image regions that do or do not have an acceptable planar approximation. Furthermore, it is clearly desirable to be able to distinguish between acceptable approximations by optimising a functional taking into account the closeness of the approximation. Modelling the image I as a piecewise constant function with added Gaussian noise, the maximum likelihood criterion is to minimise the total variance

$$\mathrm{Var} = \sum_{i=1}^{r} \sum_{p \in X_i} (I(p) - \mu_i)^2$$

where $X_1, \ldots, X_r$ is a partition of the image into connected regions and where μ_i is the mean value of pixels in region X_i. To simultaneously minimise the number of regions r, we can minimise $\mathrm{Var} + \lambda r$ for some constant λ. This leads to the following definition.

Definition 2.4. An *optimal segmentation* of an image is a partition of the image into connected regions $X_1, \ldots, X_r$ which minimises $\mathrm{Var} + \lambda r$.

Note that an optimal segmentation does not necessarily satisfy Definition 2.2 of a segmentation for any given uniformity predicate U. It is clear that Definition 2.4 can be generalised by approximating the image by any piecewise smooth functions, such as polynomials or splines, instead of constant functions. The simplicity criterion can be generalised to include not only the number r of regions, but also, for example, the smoothness and the length of region boundaries.

3. Polyomial Segmentation Algorithms

For one-dimensional images, a simple greedy algorithm is sufficient to find a minimal segmentation in polynomial time. Given an image $I(1), \ldots, I(N)$ we first find the largest value of $i \leq N$ such that

$$U(\{1, 2, \ldots, i\}) = \text{true}$$

Then, if $i < N$, we apply recursively the same algorithm to the remaining image $I(i+1), \ldots, I(N)$. For example, the result of this algorithm on the one-dimensional image of Fig. 2(a) is given in Fig. 2(c).

Let N_U be the number of evaluations of the uniformity predicate U by this greedy algorithm. N_U is clearly a linear function of N, the length of the image.

Let S_G represent the segmentation found by this greedy algorithm. We will show that S_G is minimal. Suppose, on the contrary, that S_M is a minimal segmentation composed of fewer regions than S_G. Suppose that the regions in S_G end at pixels $i_1, \ldots, i_r$ and the regions in S_M end at pixels $j_1, \ldots, j_s$, where $s < r$. Since S_M is composed of fewer regions than S_G, it must contain a boundary j_k which is further to the right than the corresponding boundary i_k. Let k be the first index for which $j_k > i_k$. Then the region R_G ending at i_k in S_G must be a proper subset of the region R_M ending at j_k in S_M. This is illustrated in Fig. 2(d). Furthermore, $R_G \cup \{i_k + 1\} \subseteq R_M$. By the subset property, $U(R_G \cup \{i_k + 1\}) = U(R_M) =$ true, which contradicts the choice of i_k by the greedy algorithm.

An obvious question is whether an optimal segmentation of a one-dimensional image, according to Definition 2.4, can also be found in polynomial time. In fact, it is possible to find an optimal segmentation in quadratic time, using a dynamic programming algorithm.

To see this, for each $i \in [0, N]$, let $S_{\text{OPT}}(i)$ represent the optimal segmentation of the truncated image $I(1), \ldots, I(i)$ and let Score(i) be its score given by $\text{Var} + \lambda r$. If the last region in $S_{\text{OPT}}(i)$ is $[k+1, i]$, then

$$\text{Score}(i) = \text{Score}(k) + \text{Var}(\{I(k+1), \ldots, I(i)\}) + \lambda$$

where

$$\begin{aligned} \text{Var}(\{I(k+1), \ldots, I(i)\}) &= \sum_{t=k+1}^{i} (I(t) - \mu)^2 \\ &= \sum_{t=k+1}^{i} I(t)^2 - (i-k)\mu^2 \end{aligned}$$

with $\mu = \frac{1}{i-k} \sum_{t=k+1}^{i} I(t)$.

Thus, $S_{\text{OPT}}(i)$ is given by

$$S_{\text{OPT}}(i) = S_{\text{OPT}}(k) \cup \{[k+1, i]\}$$

where $k \in [0, i-1]$ is such that

$$\text{Score}(k) + \text{Var}(\{I(k+1), \ldots, I(i)\}) + \lambda$$

is minimal. We can now use a standard dynamic programming algorithm to determine $S_{\text{OPT}}(N)$ in $O(N^2)$ time and $O(N)$ space.

It is clear that the one-dimensional case is much simpler than the two-dimensional problem. A single, not necessarily minimal or optimal, segmentation in two dimensions can be found in polynomial time, in fact using only a linear number of evaluations of the uniformity predicate, by the following merging algorithm.

We maintain an index $\text{REG}(p) =$ region containing the pixel p.

```
SEG_BY_MERGING:
  {First segment the image into individual pixels}
  for all pixels p do REG(p) := {p};

  for all pairs of neighbouring pixels p, q do
      if REG(p) and REG(q) are not the same region
      then if REG(p) ∪ REG(q) satisfies U
           then merge REG(p) and REG(q)
```

The order in which the pairs of pixels (p, q) are examined is unspecified. Different orders may produce different segmentations.

SEG_BY_MERGING grows but never shrinks regions. If at some point $\text{REG}(p) \cup \text{REG}(q)$ does not satisfy U, then we know, by the subset property, that $R \cup S$ cannot satisfy U, for any pair of regions $R \supseteq \text{REG}(p)$, $S \supseteq \text{REG}(q)$. This shows that, for every pair of pixels p, q, there is no need to test twice whether $\text{REG}(p) \cup \text{REG}(q)$ satisfies U, since any growing of REG(p) or REG(q) cannot change the value of $U(\text{REG}(p) \cup \text{REG}(q))$ from false to true.

SEG_BY_MERGING clearly makes $O(n)$ evaluations of the uniformity predicate U, where n is the number of pixels in the image. Merging two regions requires updating REG(r) for all pixels r in one of the two regions. Despite this, the total number of updates is bounded above by $n \log_2 n$ provided that we always update the pixels in the smaller of the two regions. Indeed, for each pixel r, the index REG(r) will be updated a maximum of $\log_2 n$ times: each time REG(r) is updated, the size of the region REG(r) at least doubles, and the maximum size of REG(r) is bounded above by n, the number of pixels in the image. The time complexity of SEG_BY_MERGING is thus $O(nc + n \log n)$ where c is the cost of testing whether $\text{REG}(p) \cup \text{REG}(q)$ satisfies U.

Split-and-merge algorithms, which start with an initial segmentation into regions of size k pixels, are not only more robust to noise but have also been found in practice to be more efficient by a constant factor (Pavlidas, 1977). Our intention in presenting SEG_BY_MERGING is simply to show formally that an efficient algorithm to find a single segmentation does exist.

4. Minimal Segmentation in Two Dimensions

In this section we show that finding a valid 2D segmentation with the minimum number of regions is NP-complete. We will then use the same construction to prove the NP-completeness of finding an optimal segmentation (according to Definition 2.4) of a two-dimensional image. Since complexity theory (Garey and Johnson, 1979; Papadimitriou, 1994) only applies to decision problems, we express minimal two-dimensional segmentation as a decision problem.

MIN2DSEG: Given an $N \times N$ image I, a uniformity predicate U and a constant k, does there exist a valid segmentation S of I using this predicate U such that S has at most k regions?

We will show the NP-completeness of MIN2DSEG in the case that the uniformity predicate is given by (1), above. Other uniformity predicates may give rise to problems which are polynomially solvable.

There is a polynomial equivalence between the decision problem MIN2DSEG and the problem of determining the minimal number of regions in all valid segmentations of I, since the latter problem is equivalent to solving MIN2DSEG for each of $k = 1, 2, \ldots, N^2$.

It is clear that MIN2DSEG is in NP, since we can verify in polynomial time that a given segmentation is valid and consists of at most k regions. To prove that MIN2DSEG is NP-complete we give a polynomial reduction from PLANAR 3SAT to MIN2DSEG. Since PLANAR 3SAT is NP-complete (Lichtenstein, 1982), any polynomial-time algorithm for MIN2DSEG would thus provide polynomial-time algorithms for all problems in NP.

An instance of 3SAT is a set of Boolean variables $\{v_i\}$ and a set of disjunctions of 3 literals $\{D_j\}$. A solution is an assignment of truth values to the variables $\{v_i\}$ such that all the disjunctions are simultaneously satisfied. An instance of PLANAR 3SAT is an instance of 3SAT that can be represented by a planar graph G. There is a vertex in G for each variable v_i and for each disjunction D_j; there is an edge linking v_i and D_j if and only if v_i or $\neg v_i$ is one of the three literals in D_j.

Theorem 4.1. *MIN2DSEG is NP-complete.*

Proof: The theorem follows from the following polynomial reduction of PLANAR 3SAT to MIN2DSEG.

Given an arbitrary instance of PLANAR 3SAT, we will construct an image I and a value k such that I has a valid segmentation consisting of at most k regions if and only if this instance of PLANAR 3SAT is satisfiable.

The image I is built from three types of constructions: one for each variable v_i, one for each disjunction D_j and one to negate a variable v_i to produce $\neg v_i$.

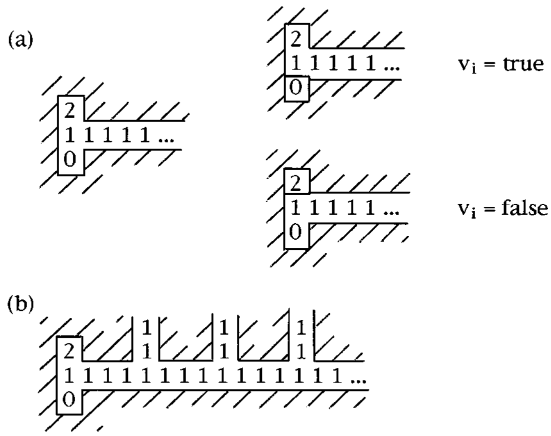

Figure 3. (a) A variable construction and its two possible minimal segmentations, (b) the branching out of a variable construction towards several other constructions.

We choose the uniformity predicate to be

$$U(X) = \text{true} \quad \text{iff} \quad \text{for all points } p, q \in X \quad |I(p) - I(q)| \leq 1$$

In other words, a region is uniform if its pixel values lie within 1 of each other.

The construction representing the variable v_i is shown on the left of Fig. 3(a). The shaded area in all figures represents a background area with a constant pixel value of 4. Such a value guarantees that the background is segmented independently of the foreground, and can thus be ignored, since the foreground only contains pixels of value 0, 1 or 2. The construction on the left of Fig. 3(a) has only two minimal segmentations (shown on the right of Fig. 3(a)) which we associate, as indicated, with the two possibilities $v_i = \text{true}$ and $v_i = \text{false}$. The segmentation corresponding to $v_i = \text{true}$ retains the value 2 whose influence is propagated to all disjunctions D_j containing v_i. Figure 3(b) shows how a single construction of the type shown in Fig. 3(a) can fork out towards many disjunctions.

On the left of Fig. 4(a) is the construction to negate a variable. The input region at the left is connected to the construction for variable v. The critical pixels are those of value 1 which have neighbouring pixels of value 0 and 2; they can be in the same region as either the 0's or the 2's. There are only two minimal segmentations and these are shown on the right of Fig. 4(a). If $v = \text{false}$, then the output region at the right of the construction contains a 2, which by convention corresponds to the value true. If $v = \text{true}$ then the output region contains a 0 corresponding to the value false.

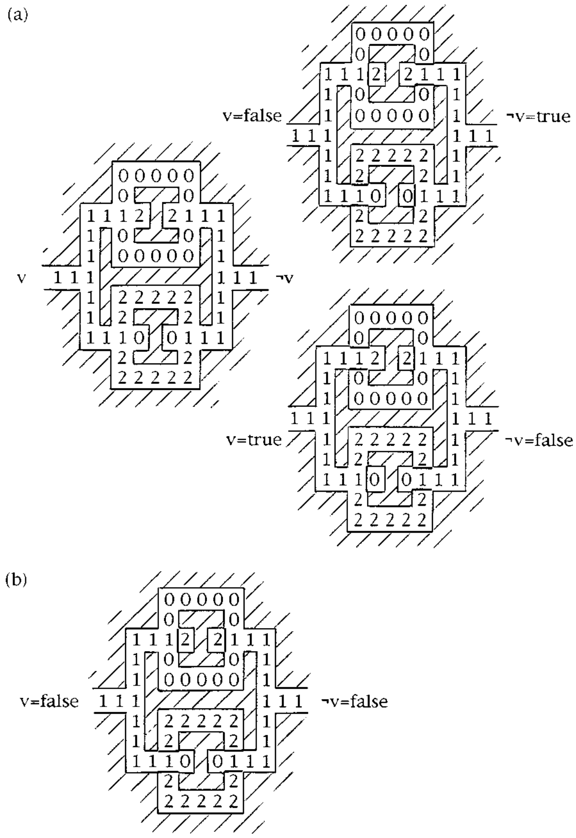

Figure 4. (a) A negation construction and its two possible minimal segmentations, and (b) an example of a non-minimal segmentation.

Forcing the input and output regions to both contain the value 0, for example, produces a non-minimal segmentation as illustrated in Fig. 4(b).

The construction for the disjunction $v_1 \vee v_2 \vee v_3$ is shown in Fig. 5. (In fact, eight copies are shown.) The construction is connected at the top to the constructions (of the type shown in Fig. 3(a)) for the three variables v_1, v_2, v_3 as indicated. The construction contains two copies of the negation construction of Fig. 4(a). The shaded area has a pixel value of 4 and the white "paths" have a pixel value of 1, except for a single 2, as shown. There are seven minimal segmentations corresponding to the seven combinations of values of v_1, v_2, v_3 which satisfy $v_1 \vee v_2 \vee v_3$. All eight cases are shown, corresponding to the eight possible assignments of truth values to v_1, v_2, v_3. If $v_1 = v_2 = v_3 =$ false, then the best we can do is the segmentation shown at the bottom right of Fig. 5 which has one more region than the other seven minimal segmentations shown in Fig. 5.

We now have all the necessary constructions to reduce a graph G corresponding to an instance of

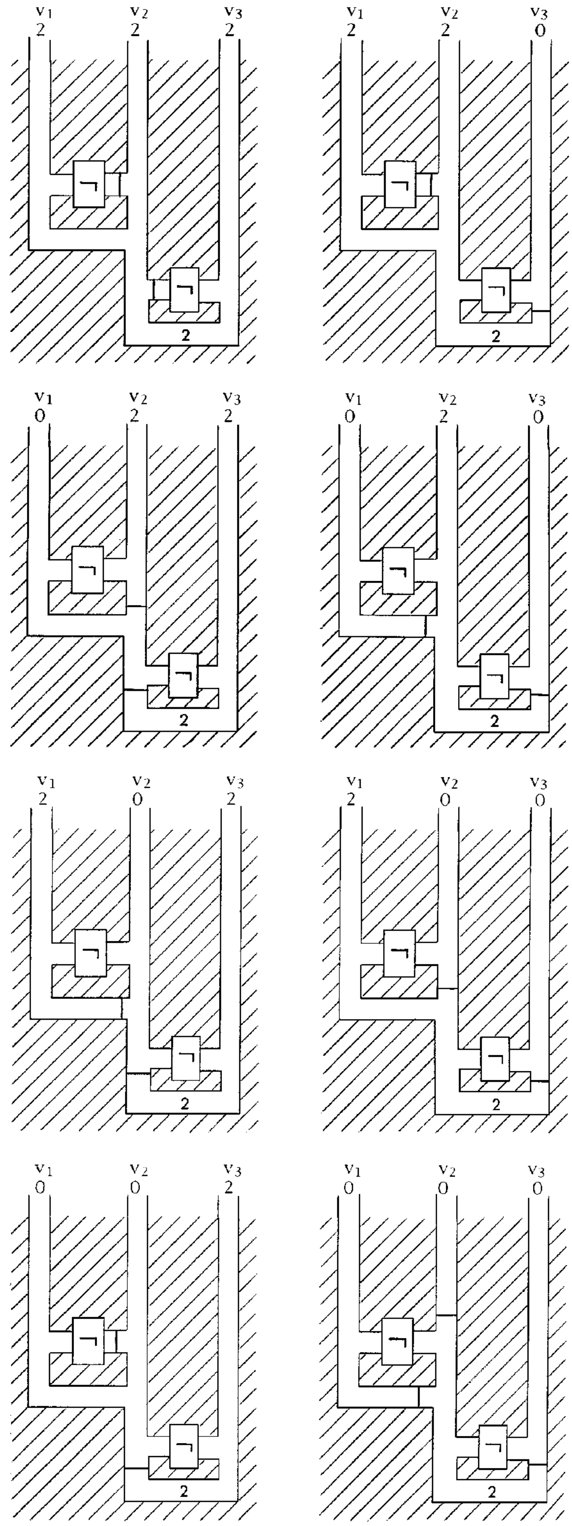

Figure 5. The disjunction construction for $v_1 \vee v_2 \vee v_3$ and examples of minimal segmentations for each of the eight combinations of values for v_1, v_2, v_3.

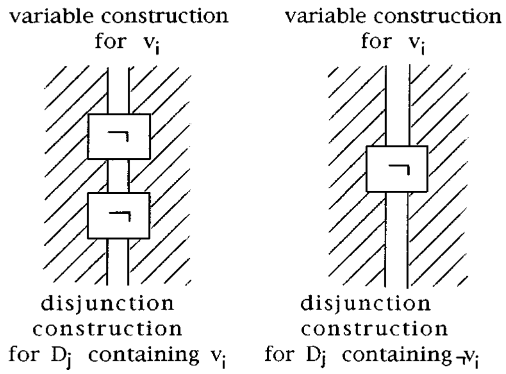

Figure 6. (a) Two negation constructions separate the variable construction for v_i and the disjunction construction for D_j containing v_i, and (b) a single negation construction separates the variable construction for v_i and the disjunction construction for D_j containing $\neg v_i$.

PLANAR 3SAT into an image to be segmented. Each vertex of G corresponding to a variable v_i is transformed into a variable construction, as illustrated in Fig. 3(a). Each vertex corresponding to a disjunction D_j is transformed into a disjunction construction, as illustrated in Fig. 5. For each negated variable $\neg v_i$ in D_j, we add a negation construction, as illustrated in Fig. 4(a), to the input region corresponding to $\neg v_i$ in the disjunction construction for D_j. For each non-negated variable v_i in D_j, we add two negation constructions, to guarantee independence between different disjunction constructions. This double negation does not alter the logical value of v_i. This is illustrated in Fig. 6.

Let b denote the number of background regions in the image, n the number of Boolean variables v_i, m the number of disjunctions D_j and p the number of negation constructions (not counting those within disjunction constructions). Define the constant k as

$$k = b + 2n + 4m + 3p$$

Each variable construction adds at least two regions to the total number of regions. Each disjunction construction adds at least four regions to the total number of regions. Each negation construction adds at least three regions to the total number of regions. The constructed image has a valid segmentation of k regions only if each variable construction has a minimal segmentation, as illustrated in Fig. 3(a), each negation construction has a minimal segmentation, as illustrated in Fig. 4(a), and each disjunction construction has a minimal segmentation, as illustrated in Fig. 5.

The assignment of truth values corresponding to such a segmentation satisfies each of the disjunctions D_j and is hence a solution to the instance of PLANAR 3SAT. Conversely, from a solution for the instance of PLANAR 3SAT, we can construct a segmentation of the image consisting of exactly k regions.

Since this reduction from PLANAR 3SAT to MIN2DSEG is clearly polynomial, we can deduce that MIN2DSEG is NP-complete. □

The above proof uses $d = 5$ brightness levels. It is an open problem whether MIN2DSEG remains NP-complete for $d = 3$ or 4. MIN2DSEG is trivially solvable in polynomial time for $d = 2$, since there is only one non-trivial segmentation. The above proof makes use of a uniformity predicate which specifies that no two pixel values differ by more than a given constant value. Simpler uniformity predicates, such as thresholding at fixed levels, uniquely determine a segmentation, and hence the minimal segmentation can again be found in polynomial time (Pavlidas, 1977).

In MIN2DSEG the fitting of a piecewise planar function to the image is achieved using a uniformity predicate and the only optimisation criterion is the minimisation of the number of regions. In the variational approach, embodied in OPT2DSEG below, function fitting and the minimisation of the number of regions are carried out simultaneously by the optimisation of a single combined functional.

OPT2DSEG: Given an $N \times N$ image I and constants λ, c, does there exist a partition of the image into connected regions $X_1, \ldots, X_r$ such that $\mathrm{Var} + \lambda r \leq c$?

Theorem 4.2. *OPT2DSEG is NP-complete.*

Proof: We can use the same reduction from PLANAR 3SAT as in the proof of Theorem 4.1, except that, because of the new optimisation criterion $\mathrm{Var} + \lambda r$, the constructions need to be slightly modified to avoid spurious alternative segmentations. It is necessary to make the constructions vary as a function of λ, otherwise for very small values of λ the optimal segmentation would be an oversegmentation into many regions of zero variance and for very large values of λ the optimal segmentation would be an undersegmentation comprising a single region of high variance.

The negation construction of Fig. 4(a) is stretched in a horizontal direction so that the single-pixel areas of value 0 or 2 now contain $\lceil \lambda/4 \rceil$ pixels. The constructions in Figs. 3(a) and 5 are similarly modified so that the single-pixel areas of value 0 or 2 now contain $5\lceil\lambda/4\rceil + 10$ pixels. Double negation constructions, as shown on the left-hand side of Fig. 6, are added in order to limit the size of regions of pixel value 1 to a maximum of 19 pixels.

With these modifications, for the class of images constructed in the proof of Theorem 4.1, the optimal segmentation is identical to the minimal segmentation for all values of $\lambda \geq 3$. □

5. Discussion of the Tractability of Segmentation

To put into perspective the importance of Theorems 4.1 and 4.2, we should remember that NP-completeness results only prove that no algorithm exists which has polynomial worst case time complexity (assuming $P \neq NP$). They tell us nothing about the existence of either *heuristics* with worst case polynomial time complexity or algorithms with *average case* polynomial time complexity. (See Tsotsos (1990) and Kube (1991) for a more detailed discussion of the relevance of applying computational complexity results to visual problems.)

We should also ask ourselves whether the segmentation problems studied in this paper are the most appropriate abstractions of the diverse image interpretation problems encountered in practice. Two fundamental assumptions about images and the scenes they represent are hidden in the definitions of minimal and optimal segmentations:

1. semantic entities (such as objects) tend to have a uniform aspect, implying that regions should satisfy the uniformity predicate or can be approximated by a constant function.
2. an interpretation involving fewer semantic entities, and hence fewer regions, is a priori more likely.

Segmentation is an exclusively region-based approach, a philosophy that we have tried to preserve in this paper. However, it is also possible to incorporate in the optimisation function other criteria concerning, for example, the boundaries between regions. These could be based on fundamental assumptions, such as the assumption that object boundaries are more likely to be smooth and to represent an important difference between pixel values on either side of the boundary.

It should be noted that the condition that regions be connected is only one possible criterion for region coherence. In scene analysis a partially occluded object is often visible in two or more unconnected areas of the image. It is possible to imagine alternative criteria for region coherence, based on, for example, the boundary shape of the union of possibly unconnected areas.

It is, of course, very doubtful whether generalising the problem of optimal segmentation in any of these ways would render it solvable in polynomial time. An alternative optimisation criterion is to minimise the total length of region boundaries in the segmentation. However, Eades and Rappaport (1993) have proved the NP-hardness of the related problem of finding a minimal separating polygon for two given finite sets of points in the plane. A separating polygon which separates S from T is a simple polygonal circuit P such that every point of S lies inside or on the boundary of P while every point of T lies outside P. A minimal separating polygon is such that the sum of the lengths of the edges of P is minimal.

An obvious direction for future research is whether relaxing the definition of an optimal segmentation will produce a tractable problem, while keeping the essential characteristics of the problem. For example, as a compromise between minimising a global function such as the number of regions (an NP-complete problem) and having no optimisation function (a problem solvable in polynomial time), is to apply a greedy algorithm which finds the most likely region, according to a local criterion such as maximising its size, subtracts out of the image the pixels which belong to this region, and recursively calls the same algorithm on the remaining image. The largest possible uniform region can be determined in polynomial time for simple uniformity predicates of the form

$$U(X) = \text{true} \quad \text{iff} \quad \text{for all points } p, q \in X \quad |I(p) - I(q)| \leq k$$

by thresholding and merging operations.

6. Model-Based Segmentation

The segmentation of an image using a library of object models is often known as scene analysis. In fact, the problem is not only to segment the image but also to recognise and locate objects in a 3D scene from the information contained in its projection into a 2D image. The loss of information in passing from the 3D scene to the 2D image is compensated by knowledge of some or all of the objects which may occur in the scene.

Compared with the recognition of isolated objects, the interpretation of multi-object images presents two extra difficulties:

1. greater ambiguity in the recognition of individual objects due to partial occlusion by nearer objects or contrast failure with objects projecting into adjacent regions.
2. possible combinatorial ambiguity due to the presence of many objects, none of which can be recognised unambiguously.

Most workers have only concerned themselves with the first and most immediate of these two problems. It is known that the recognition of a partially visible object independently of the rest of the scene is essentially a polynomial-time problem (Cass, 1992). Examples of polynomial-time techniques that have been successfully employed are hypothesise and test (Ayache and Faugeras, 1986), hashing (Lamdan et al., 1988), clustering in transformation space (Cass, 1992) and multi-resolution (Neveu et al., 1986). These methods assume that the partially visible object is occluded by unknown objects. If all the occluding objects can be reliably identified and their projections subtracted out of the image, then this can only increase the reliability of the recognition of the partially occluded object (Cooper, 1988; Ullmann, 1992). Unfortunately, when there is ambiguity, both in the correct segmentation of the image and in the semantic labelling of regions, this leads to a combinatorial problem which in the most general case is intractable, as we will demonstrate in the following section.

In order to analyse the complexity of scene analysis, we need a formal definition of a valid interpretation, which we call a reconstruction. Many constraints can be applied in scene analysis based on knowledge of physical laws such as the law of gravity or on high-level knowledge of the possible positions of different objects (such as blackboards may be attached to walls but never to ceilings). Most collections of objects in 3D space are highly unlikely or physically impossible. We restrict ourselves to applying the most basic physical constraint, that two solid objects cannot occupy the same space.

Definition 6.1. Given a set M of 3D object models and a set T of 3D transformations, a *physically possible scene* is a set S of transformed objects $t(m)$ ($m \in M$ and $t \in T$), such that for all $t_1(m_1), t_2(m_2) \in S$,

$$t_1(m_1) \cap t_2(m_2) = \emptyset$$

Definition 6.2. Given an image I, a projection transformation P, a set M of 3D object models and a set T of 3D transformations, a *reconstruction* of the image I is a set $R \subseteq M \times T$ such that

$$S = \{t(m) : (m, t) \in R\}$$

is a physically possible scene which projects into I, i.e.,

$$P(S) = I$$

and the projection of each element of S is visible in at least one pixel of the image I.

We can now state the decision problem corresponding to the reconstruction of a 3D scene from a 2D image.

RECONSTRUCTION: Given an image I, a projection transformation P, a set M of 3D object models and a set T of 3D transformations, does there exist a reconstruction of I?

It is known that RECONSTRUCTION is solvable in polynomial time when scenes cannot contain cycles of mutually occluding objects and when the only depth cue in the image is occlusion. There are no depth cues, for example, if the image was produced by an orthographic projection or if the scene is a pile of flat objects. In fact, if a single reconstruction can be found in polynomial time, then a compact description of *all* reconstructions can also be determined in polynomial time (Cooper, 1992).

A sufficient condition for RECONSTRUCTION to be polynomially solvable is that the set of possible images can be generated by first calculating the set $P(T(M))$ of projections of transformed objects and then displaying elements of $P(T(M))$ one behind the other in the image array. An image generated by this mechanism can be interpreted without backtracking by finding the foremost object $P(t(m))$, subtracting $P(t(m))$ out of the image and then applying the same algorithm recursively to the remaining image (Cooper, 1988, 1992). Unfortunately, there are two reasons why this generation mechanism may be inaccurate:

(i) certain physically possible images cannot be generated, since cycles of mutually occluding objects cannot be generated by displaying one object behind another;

(ii) physically impossible images may be generated since the non-intersection of transformed objects is not tested.

It is already known that RECONSTRUCTION is an NP-complete problem when the set of possible scenes is enlarged to include cycles of mutually occluding objects. In fact, this problem even remains NP-complete when the correct segmentation of the image is known (Cooper, 1992). In the following section, we complete the analysis of this problem by showing that, even when objects cannot form cycles, RECONSTRUCTION is NP-complete when the image contains depth cues such as perspective, shadows or range values. Intractability is due to the non-intersection constraint on transformed objects. It is curious to note that it is the introduction of depth information and an extra constraint to reduce the number of possible interpretations that renders the problem intractable. We will show that if depth information can be deduced from images, then an algorithm to solve RECONSTRUCTION could also be used to solve jigsaw puzzles. We first need to show that the problem of solving jigsaw puzzles is NP-complete.

7. The Jigsaw Problem

A special case of the reconstruction of a scene occurs when we know that there is no occlusion. In other words, the objects fit together to cover the whole scene without overlapping, in the same way that the pieces of a jigsaw fit together to form a picture. This problem is of more practical importance in one dimension than in two, in applications such as speech or text analysis. Fortunately, in one dimension the problem can be solved in polynomial time using a dynamic programming algorithm.

Definition 7.1. A *jigsaw* is an image I, a set of objects M, a set of transformations T and a projection operation P.

Note that we assume that the image to be reconstructed is given. A solution to a jigsaw is a reconstruction, according to Definition 6.2, in which the projections of no two transformed objects overlap.

Definition 7.2. A *solution to a jigsaw* (I, M, T, P) is a set S of transformed objects $t(m) \in T(M)$ such that

1. for all $t_1(m_1), t_2(m_2) \in S$

$$P(t_1(m_1)) \cap P(t_1(m_1)) = \emptyset$$

2. $P(S) = I$.

In a children's jigsaw puzzle all pieces must be used exactly once. Even without this extra constraint, the problem of reconstructing a jigsaw is NP-complete in two dimensions.

Definition 7.3. A *minimal solution to a jigsaw* (I, M, T, P) is a solution of minimal cardinality.

We now state two decision problems concerning jigsaws.

JIGSAW: Does the jigsaw (I, M, T, P) have a solution?

MINIMAL JIGSAW: Does the jigsaw (I, M, T, P) have a solution of cardinality less than or equal to k?

We will show that JIGSAW is NP-complete in two dimensions, but that MINIMAL JIGSAW is solvable in polynomial time in one dimension.

When a transformed object $t(m)$ is projected into a one-dimensional image of length n, we assume that the result is connected. We denote its left-most pixel by *start*$(t(m))$ and its right-most pixel by *end*$(t(m))$. Transformed objects $t(m)$ which overlap the end of the image, of length n, are considered to end at pixel n.

Definition 7.4. A *partial jigsaw solution* is a list $t_1(m_1), \ldots, t_r(m_r)$ of transformed objects which match exactly the image on pixels $1, \ldots, i$ and which satisfy

1. for $q = 1, \ldots, r-1$, *start*$(t_{q+1}(m_{q+1})) =$ *end* $(t_q(m_q)) + 1$
2. *end*$(t_r(m_r)) = i$.

The following dynamic programming algorithm DP-JIGSAW solves MINIMAL JIGSAW in one dimension by calculating $D(i)$, for $i = 0, \ldots, n$, where

$$D(i) = \begin{cases} \text{minimum number of objects in a partial} \\ \quad \text{jigsaw solution } t_1(m_1), \ldots, t_r(m_r) \\ \quad \text{such that } t_r(m_r) \text{ ends at pixel } i \\ \infty \quad \text{if no such partial solution exists} \end{cases}$$

DP-JIGSAW:

Initialise all $D(i)$, $i = 1, \ldots, n$, to infinity and $D(0)$ to 0.

for $i := 1$ to n do
  for all transformed objects $t(m)$ ending at pixel i do
    if $t(m)$ matches the image
    then $D(i) := \min(D(i),$
      $1 + D(\max(0, i - \text{len}(t(m)))))$

$\text{len}(t(m))$ is the length of transformed object $t(m)$. The complexity of DP-JIGSAW will be dominated by the time to find all correspondences between objects and the image. These could be determined beforehand by classic linear-time pattern matching algorithms (Aho et al., 1974). A dynamic programming algorithm which allows for local distortions in the appearance of $t(m)$ in the image, by minimising the string-edit distance between $t(m)$ and the image, has been widely used to find correspondences between object and image contours (Ansari and Delp, 1990; Gupta and Malakapalli, 1990).

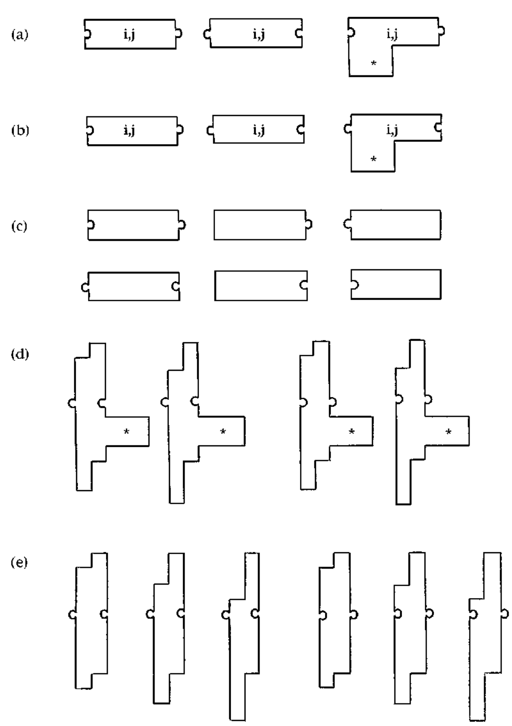

Figure 7. A set of jigsaw pieces.

Theorem 7.5. *JIGSAW is NP-complete.*

Proof: Given a putative solution S to a jigsaw (I, M, T, P), it is simple to verify in polynomial time whether S is a solution. Thus, to demonstrate NP-completeness it is sufficient to give a polynomial reduction from 3SAT to JIGSAW.

Let $D_1 \wedge \cdots \wedge D_m$ be any instance of 3SAT on n variables. We will construct a jigsaw (I, M, T, P) which has a solution iff this instance of 3SAT has a solution.

We choose T to be the set of all rigid transformations in a plane parallel to the image plane and P an orthographic projection. This implies that objects are not dilated by their projection into the image.

The set M of objects is illustrated in Fig. 7. There are two categories of objects: "horizontal" objects (Figs. 7(a), (b), and (c)) and "vertical" objects (Fig. 7(d) and (e)). For each (i, j) such that v_i is a literal in D_j, there are objects of the kind illustrated in Fig. 7(a). For each (i, j) such that $\neg v_i$ is a literal in D_j, there are objects of the kind illustrated in Fig. 7(b). The characters "i, j" are part of the surface markings of the object. Their purpose is to restrict the possible placement of the object to a unique position in the image, the position at which the same pattern "i, j" occurs in the image. Horizontal objects fall into two classes: either they have teeth which project towards the right or they have teeth which project towards the left. Teeth facing right in row i correspond to the value true for v_i; teeth facing left correspond to the value false.

When $j = n$, the tooth on the right-hand side of the six objects in Figs. 7(a) and (b) is absent. This is simply due to the fact that these objects will always occur on the very right-hand side of the jigsaw and hence do not need teeth to interlock with another object on their right.

The image is chosen so that the horizontal and vertical objects must be locked together to form a grid pattern. One such grid pattern is shown in Fig. 8 for the instance $D_1 \wedge D_2 \wedge D_3$ of 3SAT, where

$$D_1 = v_3 \vee v_4 \vee v_5$$
$$D_2 = \neg v_1 \vee v_3 \vee \neg v_4$$
$$D_3 = \neg v_2 \vee \neg v_3 \vee v_4$$

The jth column of this image corresponds to disjunction D_j, and the ith row corresponds to variable v_i.

If either v_i or $\neg v_i$ is one of the literals in D_j, then attached to the intersection of the ith row and the jth column there is an extra square, marked by an asterisk,

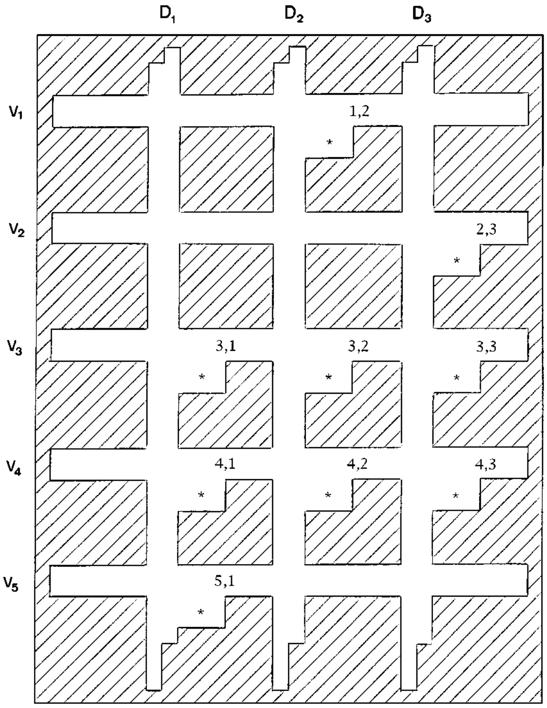

Figure 8. A jigsaw composed of a selection of the pieces illustrated in Fig. 7.

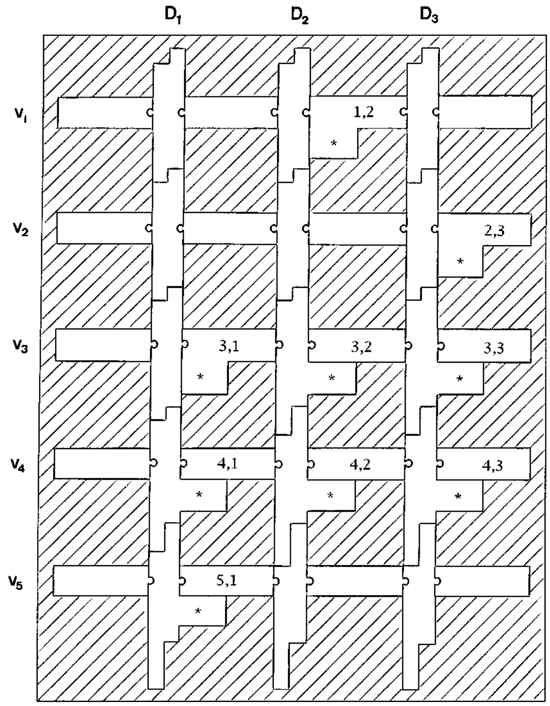

Figure 9. A solution to the jigsaw of Fig. 8.

as shown in Fig. 8. Furthermore, just above this asterisk there is the pattern "i, j", as shown.

The variable v_i is considered to be assigned the value true if all objects in row i of the solution to the jigsaw have right-facing teeth, and false if all objects in row i have left-facing teeth. The vertical objects which interlock with these horizontal objects are shown in Figs. 7(d) and (e). The direction of their teeth have been chosen to ensure that objects in a given row i face either all left or all right. This conserves the truth value of v_i throughout the row.

The purpose of the vertical objects in column j is not only to propagate the same truth value along the length of each row, but also to "count" the number of literals of D_j which are false. In fact, the length of the tail which hangs from the bottom of the vertical object at the intersection of row i and column j is at least $L + 1$ units where L is the number of literals v_k or $\neg v_k$ in D_j such that $k \leq i$ and such that this literal takes on the value false. Since, by design of the image, the vertical object at the intersection of row n and column j must have a tail of length 3, the maximum number of literals in D_j which are false is 2.

Figure 9 illustrates a solution to the jigsaw of Fig. 8. The corresponding assignment of truth values to $v_1, \ldots, v_n$ is $v_1 = v_2 =$ false; $v_3 = v_4 = v_5 =$ true, since rows 1 and 2 are comprised only of left-facing pieces and rows 3, 4, 5 only of right-facing pieces. Each disjunction D_j contains at least one literal which takes on the value true. Suppose that v_i (or $\neg v_i$) is the first such literal in D_j. Then the asterisk at the intersection of row i and column j is accounted for by the hiorizontal object on the right of Fig. 7(a) (or Fig. 7(b)).

The other two asterisks in column j, corresponding to the other two literals of D_j, are accounted for by vertical objects shown in Fig. 7(d). These vertical objects have a tail which is longer by one unit compared with the length of the tail of the vertical object just above it in the same column. All other vertical objects have a tail of the same length as the vertical object just above it. At most two vertical objects of the kind shown in Fig. 7(d) can occur in the same column j, since the length of the tail can only increase from 1 to 3 units within column j. This forces at least one of the literals in D_j to be true so that at least one of the three asterisks in column j can be accounted for by a horizontal object.

The shaded area of the image is accounted for by copies of shaded objects not actually shown in Fig. 7 but which are also members of the object set. These "filler" objects are of a different colour to the horizontal and vertical objects which interlock to form the grid pattern. They are such that the shaded area can always be constructed from them.

A solution to the jigsaw constructed as above from an instance $D_1 \wedge \cdots \wedge D_m$ of 3SAT provides a solution to this instance of 3SAT. Conversely, given a solution to this instance of 3SAT, we can construct a solution to the corresponding jigsaw. □

The above NP-completeness proof of JIGSAW can be adapted so that the image and all the objects are a uniform colour. The surface markings "i, j" are replaced by unique teeth patterns. Thus, the hardness of the problem is not due to the correspondence between objects and image, but rather due to the constraint that pieces interlock perfectly. This shows that JIGSAW is NP-complete even when the image is not available.

Theorem 7.5 seems to contradict the fact that jigsaw puzzles of thousands of pieces can be completed in a few hours by a child. Jigsaw puzzles are specially designed so that they can be solved without backtracking. In a well-designed jigsaw, when two pieces successfully interlock and there is continuity in the picture along their join, we can be sure that these two pieces really do fit together.

The proof of Therem 7.5 can easily be adapted to prove the following result.

Theorem 7.6. *RECONSTRUCTION is NP-complete even when scenes cannot contain cycles of mutually occluding objects.*

Proof: We choose P to be any perspective projection which is non-orthographic. It is sufficient to employ the same polynomial transformation from 3SAT as in the proof of Theorem 7.5. The objects are assumed to be of non-negligible depth and to have a surface marking, such as a chessboard pattern, which allows us to determine their distance from the camera given knowledge of the perspective projection P. This depth information constrains the transformed objects to interlock to form a solution to the jigsaw.

Any polynomial-time algorithm for RECONSTRUCTION could thus be used to solve 3SAT. □

Theorem 7.6 holds whenever we have a source of depth information, other than occlusion. This depth information may be directly available, for example if I is a range image, or may be deduced indirectly; for example from knowledge of the perspective projection and the positions of light sources together with analysis of shadows in the image.

The motivation for minimising the number of regions in a segmentation was that an interpretation consisting of fewer objects is more likely and that fewer objects in the scene generally implies fewer regions in the image. If we have models of all the objects which may occur in the scene, then it is possible to minimise the number of objects in the interpretation rather than the number of regions.

Unfortunately, knowledge of all objects which may be visible in the image is not sufficient to guarantee being able to find the optimal segmentation in polynomial time. Indeed, the problem of minimising the number of objects in an interpretation is NP-complete. We will show that this is true even without the depth information which makes RECONSTRUCTION NP-complete.

Definition 7.7. Given an image I, a projection transformation P, a set M of 3D object models and a set T of 3D transformations, a *minimal reconstruction* of the image I is a reconstruction of minimal cardinality among valid reconstructions of I.

MINRECONSTRUCTION: Given an image I, a projection transformation P, a set M of 3D object models, a set T of 3D transformations and a constant k, does there exist a reconstruction of I of cardinality at most k?

Theorem 7.8. *MINRECONSTRUCTION is NP-complete even when transformed objects cannot form cycles and the image is formed by an orthographic projection.*

Proof: Firstly, observe that MINRECONSTRUCTION is in NP, since we can verify in polynomial time whether a putative reconstruction is valid and consists of at most k elements.

To complete the proof it is sufficient to consider the reconstruction of the class of images described in the proof of Theorem 7.5, each image representing an instance of 3SAT, and to choose k to be $2mn + n + b$, where m is the number of disjunctions, n the number of variables and b the number of objects required to fill the background. The solutions to this class of jigsaw problems are exactly the minimal reconstructions of the corresponding images. An image may have many reconstructions involving overlapping objects, but by choice of the objects and the images, all these alternative reconstructions have cardinality greater than k. Thus, any polynomial-time algorithm for MINRECONSTRUCTION could be used to solve 3SAT in polynomial time. □

8. Conclusion

The theoretical results presented in this paper can be considered as a step towards determining the boundary between those visual tasks for which reliable and efficient algorithms exist and those visual tasks for which such algorithms are impossible.

A segmentation of a two-dimensional image can be found in polynomial time. A segmentation involving the minimum number of regions can be found in polynomial time for a one-dimensional image, but this problem is NP-complete for two-dimensional images even when pixels can take on only five different brightness values. An alternative formulation of the same problem of approximating an image by a piecewise constant function while minimising the number of regions, using a variational approach, has also been shown to be solvable in polynomial time in one dimension but NP-complete in two dimensions.

A sufficient condition for the reconstruction of a 3D scene from a 2D image to be solvable in polynomial time is that no range information can be deduced from the image and that the set of objects which may occur in the scene are all known and cannot project into cycles of mutually occluding objects in the image. Both the no range information condition and the no cycles condition are necessary, since otherwise the problem is NP-complete. Minimising the number of objects in an interpretation also renders the problem NP-complete.

Solving a jigsaw puzzle is NP-complete in two dimensions, but a solution to a one-dimensional jigsaw puzzle which employs the minimum number of pieces can be found in polynomial time.

These results amply demonstrate that not all combinatorial problems are intractable. Indeed, intractability results can often point us in the right direction to find interesting versions of the same problems which are solvable in polynomial time.

References

Aho, A.V., Hopcroft, J.E., and Ullman, J.D., 1974. *The Design and Analysis of Computer Algorithms*. Addison-Wesley Publishing Company.

Ansari, N. and Delp, E.J., 1990. Partial shape recognition: A landmark-based approach. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12(5):470–483.

Ayache, N. and Faugeras, O.D. 1986. HYPER: A new-approach for the recognition and positioning of two-dimensional objects. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(1):44–54.

Beaulieu, J.-M. and Goldberg, M. 1989. Hierarchy in picture segmentation: A stepwise optimization approach. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11(2):150–163.

Borisenko, V.I., Zlatopol'skii, A.A., and Muchnik, I.B. 1987. Image segmentation (state-of-the-art survey). *Automation and Remote Control*, 48(7):837–879.

Cass, T.A. 1992. Polynomial-time object recognition in the presence of clutter, occlusion, and uncertainty. In *Proc. 2nd European Conference on Computer Vision*, Springer-Verlag, pp. 834–842.

Clowes, M.B. 1971. On seeing things. *Artificial Intelligence*, 2:79–116.

Cooper, M.C. 1988. Efficient systematic analysis of occlusion. *Pattern Recognition Letters*, 7:259–264.

Cooper, M.C. 1992. *Visual Occlusion and the Interpretation of Ambiguous Pictures*. Ellis Horwood: Chichester, UK.

Cooper, M.C. 1997. Interpreting line drawings of curved objects with tangential edges and surfaces. *Image and Vision Computing*, 15:263–276.

Cooper, M.C., Cohen, D.A., and Jeavons, P.G. 1994. Characterising tractable constraints. *Artificial Intelligence*, 65:347–361.

Dendris, N.D., Kalafatis, I.A., and Kirousis, L.M. 1994. An efficient parallel algorithm for geometrically characterising drawings of a class of 3-D objects. *Journal of Mathematical Imaging and Vision*, 4:375–387.

Eades, P. and Rappaport, D. 1993. The complexity of computing minimum separating polygons. *Pattern Recognition Letters*, 14(9):715–718.

Fu, K.S. and Mui, J.K. 1981. A survey on image segmentation. *Pattern Recognition*, 13:3–16.

Garey, M.R. and Johnson, D.S. 1979. *Computers and Intractability—A Guide to the Theory of NP-Completeness*. W.H. Freeman: San Francisco.

Gupta, L. and Malakapalli, K. 1990. Robust partial shape classification using invariant breakpoints and dynamic alignment. *Pattern Recognition*, 23(10):1103–1111.

Haralick, R.M. and Shapiro, L.G. 1985. Survey: Image segmentation techniques. *Computer Vision, Graphics and Image Processing*, (29):100–132.

Huffman, D.A. 1971. Impossible objects as nonsense sentences. In *Machine Intelligence*, B. Meltzer and D. Michie (Eds.), Edinburg University Press, vol. 6, pp 295–323.

Kirousis, L.M. 1990. Effectively labeling planar projections of polyhedra. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(2):123–130.

Kirousis, L.M. 1993. Fast parallel constraint satisfaction. *Artificial Intelligence*, 64:147–160.

Kirousis, L.M. and Papadimitriou, C.H. 1988. The complexity of recognising polyhedral scenes. *Journal of Computer and System Science*, 37(1):14–38.

Kube, P.R. 1991. Unbounded visual search is not both biologically plausible and NP-Complete. *Behavioral and Brain Sciences*, 4:768–770.

Lamdan, Y., Schwartz, J.T., and Wolfson, H.J. 1988. Object recognition by affine invariant matching. In *Proc. IEEE Computer Soc. Conf. Computer Vision and Pattern Recognition*, pp. 335–344.

Lichtenstein, D. 1982. Planar formulae and their uses. *SIAM J. Computing*, 11:329–343.

Lim, Y.M. and Lee, S.U. 1990. On the color image segmentation algorithm based on the thresholding and the fuzzy c-means techniques. *Pattern Recognition*, 23(9):935–952.

Mumford, D. and Shah, J. 1989. Optimal approximations of piecewise smooth functions and associated variational problems. *Communications in Pure and Applied Mathematics*, 42:577–685.

Neveu, C.F., Dyer, C.R., and Chin, R.T. 1986. Two-dimensional object recognition using multiresolution models. *Computer Vision, Graphics and Image Processing*, 34:52–65.

Nitzberg, M., Mumford, D., and Shiota, T. 1991. Filtering, segmentation and depth. *Lecture Notes in Computer Science*, vol. 662, Springer-Verlag.

Pal, N.R. and Pal, S.K. 1993. A review on image segmentation techniques. *Pattern Recognition*, 26:1277–1294.

Papadimitriou, C.H. 1994. *Computational Complexity*. Addison-Wesley: Reading, MA.

Parodi, P. and Torre, V. 1994. On the complexity of labeling perspective projections of polyhedral scenes. *Artificial Intelligence*, 70:239–276.

Pavlidas, T. 1977. *Structural Pattern Recognition*. Springer-Verlag.

Sugihara, K. 1986. *Machine Interpretation of Line Drawings*. MIT Press: Cambridge, MA.

Tsotsos, J.K. 1990. Analyzing vision at the complexity level. *Behavioral and Brain Sciences*, 13:423–469.

Ullmann, J.R. 1992. Analysis of 2-D occlusion by subtracting out. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(4):485–489.